REMARKS

Claims 6 and 14 are amended to correct drafting errors that rendered them indefinite and uncertain and are not amended in response to the Examiner's rejection of these claims under 35 USC 103(a).

The following numbered sections respond to similarly numbered sections of the office action.

1.

Claims 1-5 and 9-13 are rejected under 35 USC 103(a) as being unpatentable over U.S. Patent 6,305,001 (Graef) in view of U.S. Patent 6,536,0214 (Hathaway). The Examiner is respectfully requested to withdraw the rejections of these claims in view of the following comments distinguishing these claims over the combination of Graef and Hathaway.

2.

The Examiner cites the following sections of Graef as teaching the content of the preamble of claims 1 and 9: abstract, col. 9, lines 11-32, col. 10, lines 11-28, col. 11, lines 6-9, xol. 12, lines 6-56, FIG. 6,FIG. 7, FIG. 9, and FIG. 10. Since the scope of a claim 1 or 9 is mainly defined by the elements of the claim, rather than its preamble, the question as to whether Graef teaches the content of the preamble of claim 1 or 9 may not be particularly relevant to the issue of the patentability of those claims. Nonetheless, the applicant will respond to the Examiner's assertion that the above sections of Graef teach the content of the preamble of claims 1 and 9 to point out some concepts that will be of value when considering whether Graef teaches the elements of claims 1 and 9.

The preambles of claims 1 and 9 relate to a method or a program for synthesizing a clock tree for an IC layout, and the cited sections of Graef do relate to a method for synthesizing a clock tree. However, the preambles of claims 1 and 9 indicate that the invention relates to a method or a program for synthesizing a clock tree for a particular kind of IC layout -- a layout that has been organized into a plurality of base level partitions and a top level partition, wherein each partition occupies a separate area of the IC's semiconductor substrate. The cited sections of Graef do not talk about a clock tree synthesis method relating to an IC layout that has been partitioned. Graef's m thod does involve a type of "partitioning", but it is a different kind of partitioning, carried out at a different stage of the design process and for a different purpose. To understand the difference between the partitioning recited in the preamble of claims 1 and 9 and the f partitioning Graef discusses, it is helpful to have a basic understanding of the IC design process.

An IC designer initially creates a high-level "technology-independent" description ("netlist") of an IC by listing its input, internal and output signals conveyed by nets within the IC and describing the Boolean logic by which the signals are to be related. The designer then uses various CAD tools to convert the high level netlist into a "technology-dependant" (gate-level) IC netlist design that lists the particular "cells" (gates, transistors and other standard components) that are to implement the Boolean logic and indicates how the cell's terminals are to be interconnected. Note that the gate level netlist only lists the cells; it does not indicate where the cells are to be placed within the IC or how the nets interconnecting them are to be routed.

The designer then employs a placement and routing tool that processes the gate level netlist to automatically generate an IC layout indicating where each cell is to be placed in the IC's semiconductor substrate and indicating how the nets interconnecting their terminals are to be routed.

After placement and routing, a clock tree synthesis (CTS) tool designs the IC's clock trees. A clock tree is a network of conductors and buffers for delivering a clock signal to each clocked device (such as a register or flip-flop) that is to be clock by the clock signal. The clock tree synthesis tool lays out the clock tree in such a way that the edges of the clock signal will arrive concurrently at all clocked devices ("syncs"). The CTS tool designs the clock tree after placement and routing because it is necessary to know the position of each sync to take into account the distances signals must travel to each sync. A CTS "balances" a clock tree by adjusting delays of various signal paths of the clock tree so that edges arrive at all synchronizes concurrently.

The placement and routing process can be time-consuming, particularly for large ICs, because the time required to lay out an IC grows geometrically with the number of cells in the IC. One trick to reduce placement and routing time is to divide the IC design into several partitions, create a floor plan allocating a separate area of the IC 's substrate to each partition, and then separately and independently carry out the placement and routing for each partition. It is much faster to independently lay out N separate portions of an IC with an average of M cells each, than to lay out an entire NxM IC as a single block.

While such partitioning of the layout can speed up the placement and routing process, it does cause a problem with respect to clock trees. As mentioned above, it is necessary to balance a clock tree so that it delivers clock signal edges concurrently to all syncs. When an IC design is partitioned for layout purposes, portions of a clock tree may extend into more than one partition. Although a P&R tool can <u>independently</u> place and route the cells and nets of each partition, a prior art clock tree synthesis tool could not independently synthesize the portion of the clock tree residing in each partition. To balance a clock tree it was necessary for the CTS tool to know the position of each sync in every partition. Thus in the past it has been necessary to assemble all of the partition layouts into a complete IC layout and then allow the CTS tool to globally synthesize a balanced clock tree for that IC. As it syntheses the clock tree, the CTS tool modifies each partition because it has to add the buffers and nets forming the clock tree to the layout.

Suppose now, that after the layout and clock tree synthesis process is complete, there is a change in the logic design of one of the partitions. We would like to be able to change only the layout of that particular partition. But since any change to the layout of any one partition can unbalance the clock tree, it is necessary to globally redesign the clock tree, and that can affect the layout of all other partitions.

The applicant's invention as recited in claims 1 and 9 solves that problem in the context of a partitioned IC layout because it allows portions ("subtrees") of a clock tree within separate partitions to be laid out and balanced independently. A change in a subtree within any one partition can be accommodated without having to alter the layout of subtrees residing within other partitions.

Graef

As mentioned above, a clock tree normally delivers clock signals edges concurrently to all syncs connected to the tree. But the design of some ICs requires various groups of the IC's syncs to be clocked at different times (i.e. with differing phases) relative to a clock signal's edges. For example when a clock signal has a period P, the IC design might specify that a group I of its syncs be clocked at the same time during each clock cycle, another group II of its syncs be clocked P/4 seconds later than group I, another group III of syncs be clocked P/4 seconds after group II, and still another group IV of syncs be clocked P/4 seconds after group III.

Graef teaches that a clock tree can be designed to deliver edges of the clock signal to separate groups of syncs different times. To do so, Graef teaches to organize the high level (technology independent) netlist into "partitioned groups" based on the clock phase of its logic. See step 120 of Graef's FIG. 6. A timing budget is then created for each partition group (step 125) assigning each to a different relative clock phase and also specifying a maximum allowable path delay between partition groups (See Graef's FIG. 7 and col. 10, lines 5-36).

At this point (FIG. 6, step 130) Graef's system synthesizes a gate-level (technology dependant) netlist (col. 12, lines 22-30) and maps each sync into the appropriate timing group (FIG. 6, step 135). After using simulation to test the circuit design at the gate level (FIG. 6, step 140), Graef's system carries out placement and routing ("physical design") (FIG. 6, step 150).

The clock tree is designed during step 150. Graef (col. 13, lines 37-65 and col. 14, line 57 though column 15, line 19) teaches that the clock tree is "segmented" into distribution groups as illustrated in FIG. 9. Group I includes all syncs that are to be clocked at some phase 0 along a subtree of the clock tree that is to deliver the clock signal to those sinks. Similarly, each group II - IV includes all syncs that are to be clocked at other phases 1-3, respectively, along with a subtree of the clock tree that is to deliver the clock signal to those syncs. The clock signal is transmitted from group-to-group with the path delays through buffers 412, 414 between groups being adjusted in accordance with the timing budget established at step 125 to ensure the proper phase relationships between groups.

Note that the subtree for each group I-III of FIG. 9 must be balanced so that it delivers clock signal edges concurrently to all syncs of that group. Note also that all groups I-III must have the same path delay. Since all groups I-III must have the same path delay, none of groups I-III can be independently balanced. That is, the CTS tool that lays out and balances the clock tree, must globally adjust the internal path delays within the subtrees of all groups I-III. Any change in path delays within group I will necessitate a change in path delays within groups II and III.

Thus the applicant's claims 1 and 9 recite a method for synthesizing a clock tree in the context of an IC design that has been partitioned, wherein each partition of the design is to be independently placed and routed in a separate area of the IC's substrate. The method claims 1 and 9 recite is concerned with solving the problem of how to coordinate timing of subtrees of a clock tree that reside in those separate areas of the substrate when those subtrees themselves have been independently synthesized. Note that the clock tree problem arises because the IC layout has been partitioned so that each partition can be laid out separately and independently. The applications solution is to:

a. Independently synthesize and balance the subtree within each partition; and then

b. Synthesize a top-level portion of the clock tree designed to compensate for differences in path delays within the subtrees residing within the separate partitions.

Thus if it becomes necessary to alter the design of one of the partitions, it is necessary to re-synthesize a balanced subtree only for that particular partition and to appropriately adjust the design of the top level portion of the clock tree if there is a change in the delay of that particular subtree. It is not necessary to alter the design of portions of the clock tree residing in any other partition.

Graef, on the other hand, teaches a method for synthesizing a clock tree in the context of an IC design in which syncs are clocked with different phases relative to a clock signal. The problem Graef wants to solve is how to clock several groups of sinks at different relative times using the same clock signal. Graef's solution is to

a. Partition the high level netlist, according to the phase with which logic in each phase, partition is to be clocked,

b. Generate a clock budget indicating a clock signal path delay required between segments of a clock tree that would supply clock signals to syncs in the separate phase partitions.

c. Create the gate level netlist

d. Place and route the entire IC,

e. Generate a balanced, segmented clock tree in which each segment serves a separate phase partition, all segments have the same path delay, path delays between segments are adjusted according to the clock budget.

Note also that the nature of the partitioning differs. In the applicant's context, the design is partitioned so that different parts of the design can be independently laid out in separate <u>areas</u> of the substrate. In Graef's system the design is partitioned according to the phase with which devices are clocked for purposes of creating a timing budget and the partitioning has nothing to do with placement and routing -- the design is placed and routed as an unpartitioned whole.

Thus, we see that Graef does <u>not</u> disclose a method for synthesizing a clock tree that operates in the context laid out by the preamble of the applicant's claims 1 and 9. Even though Graef discusses "partitioning" a design, the partitioning is done in a different way, at a different stage of the IC design process, and for an entirely different reason.

The Examiner cites the following sections of Graef as teaching the first element ("first computer instructions") of claim 1 and as teaching step a of claim 9:

Abstract,
Col. 9, lines 11-32,
Col. 10, lines 11-28,
Col. 11, lines 6-9,
Col. 12, lines 6-56,

FIG. 6
FIG. 7
FIG. 9
FIG. 10.

The first element or step involves separately synthesizing a plurality of "independently balanced subtrees". Graef fails to teach that limitation. Graef indeed syntheses a plurality of subtrees since each "clock distribution group" of Graef's FIG. 9 includes a separate subtree of a clock tree. But those subtrees are not independently balanced. Graef's teaches to create a timing budget that predetermines the phase difference in the paths between the groups before the IC layout is created. Thus when the layout is created and the clock tree synthesized, the internal delays within all of its subtrees must be uniform. Thus, for example, if in the course of balancing the subtree within group I it is necessary to increase the path delay from the group's clock input to all syncs within the group, then it is necessary to increase the path delays within all other groups II, III etc. Therefore, the balancing of subtrees within the separate groups cannot be "independent" as recited in claims 1 and 9. None of the cited sections or drawings of Graef mention anything about independently balancing subtrees. It does not appear that any part of Graef mentions anything at all about how the subtrees are balanced relative to one another.

In the applicant's claims 1 and 9, the partitions correspond to separate areas of the IC while in Graef's system partitions do not correspond to areas of an IC. Thus Graef does not teach as recited in claims 1 and 9 that each subtree of the clock tree corresponding to a separate base level partition and comprises "a start point at a perimeter of the area occupied by that base level partition." None of the cited sections or drawings of Graef mention anything about positioning start points of subtrees at points on perimeters of areas occupied by corresponding partitions since Graef's partitions don't correspond to particular areas and aren't laid out in separately identified areas. Note that Graef's FIG. 9 is a schematic diagram -- it is not a plan view of an IC. Graef provides no indication that clock distribution groups I, II, III etc are physically intermingled in the IC layout.

Thus claims 1 and 9 are patentable over Graef since Graef fails to teach or suggest separately synthesizing a plurality of independently balanced subtrees, fails to teach or suggest a subtree start points at a perimeter of an area occupied by a partition,

Hathaway

The Examiner points out that "Graef did not disclose partitioning subtrees at the layout level" and suggests that Hathaway (col. 30, line 45 - col. 31, line 33) does.

The cited section of Hathaway discusses a method for synthesizing a gated clock tree. For example Hathaway's FIG. 1a shows gated clock tree in which a set of logic gates 115, 119, 123 control whether a clock signal SCLK passes to the inputs of a set of subtrees 103-107 of a clock tree. Clock tree gating is used, for example, to turn off portions of a clock tree when not needed, thereby reducing IC power consumption. The Examiner apparently reasons that Hathaway's gated clock trees are "partitioned" by gates 103-107.

However, the applicant's invention as recited in claims 1 and 9 does not relate to the use of gates to "partition" clock trees. Claims 1 and 9 relate to a method for designing a balanced clock tree for an entire IC design that has been partitioned with each partition being placed in a separate area of a substrate. Neither Graef nor Hathaway teach methods for synthesizing clock trees in that context. Claims 1 and 9 relate to a method in which subtrees of a clock tree residing in those separate areas are independently balanced. Neither Graef nor Hathaway provide any suggestion of a clock tree synthesis method in which separate portions of a clock tree are independently balanced.

Claims 1 and 9 are therefore patentable over the combination of Graef and Hathaway.

3.

The Examiner cites various sections and figure of Graef as teaching the additional subject matter of claims 2 and 10. Claims 2 and 10 are patentable over the combination of Graef and Hathaway for reasons similar to those discussed above in connecting with their parent claims 1 and 9.

Claims 2 and 10 are further patentable over the combination of Graef and Hathaway because they recite, "at least two of the subtrees have substantially dissimilar average clock signal path delays." Graef and Hathaway do not teach that. For example, referring to Graef's FIG. 9, the subtrees within groups I-III all have the same average internal path delay even though the clock tree is design to deliver clock signal edges to syncs served by different groups at different times. The time difference is accommodated not by making the internal path delays of the subtrees differ, but by delivering the clock signal to each group I-III at a different time. Thus while the applicants invention relates to a method for balancing a clock tree when path delays through its subtrees differ, Graef teaches a method for intentionally "unbalancing" a clock tree when path delays through its subtrees are uniform.

Claims 2 and 10 are further patentable over the combination of Graef and Hathaway because they recite "path delays of paths within the top level portion of the clock tree linking the entirely noted to the start point of each subtree compensated for differences in the average path delays of the subtrees so as to substantially equalize clock signal path delays between the entry point and all syncs." Graef's "top level portion" of the subtree (i.e., buffers 410, 412 and 414) does exactly the opposite; it intentionally de-compensates for similarity in the average path delays of the subtrees so as to render signal path delay between the entry point of the clock signal (at the input of buffer 410) and the syncs served by groups I-III substantially dissimilar. See Graef, col.. 14, line 57 through col. 15, line 36.

4.

The Examiner cites Graef, col. 13, line 46-col.. 15, line 40 and FIG. 9 as teaching the additional subject matter of claims 3 and 11. These claims depend on claims 1 and 9 and are patentable over the combination of Graef and Hathaway for similar reasons. Claims 3 and 11 are further patentable over Graef and Hatchway because they recite synthesizing a balanced subtree for the top-level partition. The cited sections of Graef and Hathaway do not disclose partitioned <u>layouts</u> and therefore do not teach or suggest the recited "top level" partition or synthesizing a balanced subtree for a top level partition. Graef's FIG. 9 is a block diagram of a clock tree -- it is not a plan view of a partitioned IC layout and therefore does not show a partition IC layout. The cited section of Graef talk about synthesizing a segmented clock tree, but the clock tree segments (subtrees) do not correspond to partitions (i.e. areas) of an IC layout, they correspond to differing "timing groups" of syncs categorized according to clocking phase rather than according to the area partition of a layout in which they reside.

5.

The Examiner cites Graef, col. 13, line 46-col. 15, line 40 and FIG. 9 as teaching the additional subject matter of claims 4 and 12. These claims depend on claims 1 and 9 and are patentable over the combination of Graef and Hathaway for similar reasons. Claims 4 and 12 are further patentable over Graef and Hatchway because the recite selecting first and second base level partitions of a layout. The cited sections of Graef and Hathaway do not disclose partitioned layouts and therefore do not teach or suggest the recited "top level" partition or synthesizing a balanced subtree for a top level partition. Graef's FIG. 9 is a block diagram of a clock tree -- it is not a plan view of a partitioned IC layout and therefore does not show a partition IC layout. The cited section of Graef talk about synthesizing a segmented clock tree, but the clock tree segments (subtrees) do not correspond to partitions (i.e. areas) of an IC layout, they correspond to differing "timing groups" (col. 13, lines 50-54).

Claims 4 and 12 are further distinguished over Graef and Hathaway because they recite first and second signal paths delivering a clock signal to start points of subtrees in first and second base level partitions and having "substantially different path delays to compensate for the substantially different average clock signal path delay of the subtrees of the first and second base level partitions so that a clock signal departing the first node will arrive at each sync with the fist and second base level partitions at substantially the same time". Graef does not disclose partitions or subtrees in partitions, and does not teach providing signal paths conveying clock signals to subtrees so as to compensate for differences in average clock signal path delay of the subtrees. In Graef's system, the subtrees have th same average path delays, and Graef's signal paths (through buffers 412 and 414) provide delays that have just the opposite effect of what claims 4 and 12 recite. Their path delays are designed to make the clock signal arrive at the syncs served by subtrees 404, 406 and 408 at different times.

6.

The Examiner cites Graef, col. 3, line 39-67, col. 15, line 40 and FIG. 9 as teaching the additional subject matter of claims 5 and 15. These claims depend on claims 4 and 12 and are patentable over the combination of Graef and Hathaway for similar reasons.

7.

Claims 6-8 and 14-15 are rejected under 35 USC 103(a) as being unpatentable over Graef in view of Hathaway and U.S. Patent 6,536,024 (Minami). In view of the following comments distinguishing these claims over the cited references, the Examiner is respectfully requested to withdraw the rejection of these claims.

8.

Claims 6 and 14 (as amended) depend on claims 4 and 12 and are patentable over the combination of Graef and Hathaway for the reasons discussed above in connection with claims 4 and 12. Claims 6 and 14 further recite that path delays are adjusted by adjusting all of the following four variables:

A number of buffers included in the paths,

A size of at least one buffer included in the paths,

A position of at least one buffer included in the paths, and

A position of the selected node relative to the start points of the subtrees of the first and second partitions.

Minami teaches adjusting path delays within a clock tree by adjusting the resistance and capacitances of the conductors forming its signal paths. Minami does mention, however, in paragraph 2.3, that the path delay is influenced by positions of buffers within the signal paths. But that is just one of the four variable claims 6 and 14 recite that are all adjusted to control delays with paths supplying signals from a node to subtrees of a clock tree. Minima makes no suggestion that the number or size of buffers in the path be used as variables for controlling path delay, or that the position of the node relative to the start points of subtrees residing at the downstream ends of a clock tree path be used as a variable for controlling path delay. Minima's suggestion to adjust conductor capacitance and resistance is not relevant to the applicant's claims 6 and 14.

The Examiner has correctly taken official notice that buffer sizing with respect to controlling its drive strength relative to the number of its downstream components. However, the applicant's claims 6 and 14 do not recite adjusting buffer size for that purpose. The applicant's claims 6 and 14 recite adjusting buffer size to control the path delay through a signal path within a clock tree. That prior art may teach to adjust buffer size relative to the number of components downstream of a buffer is irrelevant to the subject matter of claims 6 and 14.

The Examiner has provided no reference and has taken no official notice that the number of buffers in a signal path of a clock tree, or the position of the starting node of that signal path relative to its end point should be used as variables for the controlling delay within a clock tree signal path. Moreover, the Examiner has not shown that any prior art method for synthesizing a clock tree has employed all four of the recited in combination for controlling a delay of a signal path within a clock tree.

9, 10.

Claims 7 and 8 depend on claim 6 and are patentable over the cited references for reasons discussed above in connection with claim 6.

In view of the foregoing amendments and remarks, it is believed that the application is in condition for allowance. Notice of Allowance is therefore respectfully requested.

Respectfully submitted,

Dal J Bedell

Daniel J. Bedell
Reg. No. 30,156

SMITH-HILL & BEDELL, P.C.
12670 NW Barnes Road, Suite 104
Portland, Oregon 97229

Tel. (503) 574-3100
Fax (503) 574-3197
Docket: SILI 2282 (01-051)

Certificate of Facsimile Transmission

I hereby certify that this paper is being facsimile transmitted to the Patent and Trademark Office on the date shown below.

Penelope Stockwell

Penelope Stockwell

12/2/03

Date